

Pour réaliser une application CRUD (Create, Read, Update, Delete) de réservation de billets de bus en PHP, voici la liste des tâches à effectuer avec des explications détaillées pour chaque objet à utiliser.

## 1. Conception de la base de données

- **Table buses** : stocke les informations de chaque bus, comme id, name, number\_plate, et route.
- **Table users** : stocke les informations des utilisateurs, comme id, name, email, password.
- **Table bookings** : gère les réservations avec des champs comme id, user\_id, bus\_id, seat\_number, booking\_date, status.
- **Table routes** : stocke les informations des trajets avec id, departure\_city, arrival\_city, departure\_time, et price.

## 2. Créer la connexion à la base de données

- Utilisez **PDO** pour établir une connexion sécurisée à la base de données.
- Créez une classe Database pour initialiser la connexion avec des méthodes pour gérer les exceptions et assurer une utilisation simple.

## 3. Création des classes modèles (Models)

- **Bus** : Représente les bus avec des méthodes pour récupérer les informations du bus (getBusDetails), créer un nouveau bus (createBus), mettre à jour les informations du bus (updateBus), et supprimer un bus (deleteBus).
- **Route** : Représente les trajets entre villes. Inclut des méthodes pour ajouter, mettre à jour, supprimer, et récupérer les informations de trajet.
- **User** : Gère les informations de l'utilisateur avec des méthodes pour inscrire un utilisateur (register), se connecter (login), et récupérer les informations de l'utilisateur.
- **Booking** : Représente une réservation de billet, avec des méthodes pour ajouter une réservation (createBooking), mettre à jour une réservation (updateBooking), annuler une réservation (cancelBooking), et récupérer les réservations d'un utilisateur (getUserBookings).

## 4. Gestion des routes avec un routeur (optionnel)

- Créez un **fichier routeur** pour gérer les routes et rediriger les requêtes vers les bonnes méthodes, par exemple en utilisant une structure switch-case.
- Définissez des routes pour chaque action, comme /create-bus, /book-ticket, etc.

## 5. CRUD pour la gestion des Bus

- **Créer un bus** : Page avec formulaire pour ajouter un nouveau bus, envoyant les données à une fonction createBus().
- **Lire les bus** : Page qui affiche la liste de tous les bus avec la méthode getAllBuses.

- **Mettre à jour un bus** : Page de modification des détails d'un bus, envoyant les données à la fonction `updateBus`.
- **Supprimer un bus** : Action pour supprimer un bus, avec la méthode `deleteBus`.

## 6. CRUD pour les trajets (Routes)

- **Créer un trajet** : Formulaire pour ajouter un trajet, par exemple entre deux villes.
- **Lire les trajets** : Liste tous les trajets disponibles avec la fonction `getAllRoutes`.
- **Mettre à jour un trajet** : Formulaire pour modifier les informations d'un trajet existant.
- **Supprimer un trajet** : Permet de supprimer un trajet, utile en cas de changement d'itinéraire.

## 7. Système de réservation (CRUD pour les Bookings)

- **Créer une réservation** : Formulaire permettant aux utilisateurs de réserver un siège, lié au bus et à la route choisis.
- **Lire les réservations** : Liste des réservations de l'utilisateur connecté, avec la méthode `getUserBookings`.
- **Mettre à jour une réservation** : Option pour changer un siège réservé ou annuler la réservation.
- **Annuler une réservation** : Méthode `cancelBooking` pour changer l'état de la réservation (par ex., de "confirmé" à "annulé").

## 8. Gestion des utilisateurs

- **Inscription et connexion** : Créez des pages d'inscription et de connexion sécurisées, utilisant le modèle `User`.
- **Profil utilisateur** : Affiche les informations de l'utilisateur, ainsi que ses réservations actives.

## 9. Interfaçage avec des formulaires HTML

- Créez des formulaires HTML pour chaque opération, avec des champs pour les détails des bus, des trajets et des réservations.
- Utilisez des boutons pour soumettre les actions CRUD et actualisez les pages avec les nouvelles données.

## 10. Sécurisation de l'application

- **Validation des entrées utilisateur** : Vérifiez et assainissez toutes les données envoyées depuis les formulaires.
- **Gestion des sessions** : Protégez l'accès aux pages de gestion des réservations avec des sessions, afin que seuls les utilisateurs authentifiés puissent réserver ou gérer leurs billets.

## 11. Test et Déploiement

- Testez chaque fonction CRUD pour vérifier le bon fonctionnement de l'application.

- **Déploiement** : Mettez l'application en ligne et configurez les accès à la base de données.

Chaque objet et classe ici simplifie l'organisation du code et centralise les fonctionnalités, rendant l'application plus modulaire et facile à maintenir.